AMENDMENTS TO THE CLAIMS

1. (Original) A significand portion of a floating point multiply accumulator (FMAC) comprising:

a multiplier receiving a first input significand, a second input significand, and a third input significand;

a propagate, kill, generate generator (PKG generator) coupled to the multiplier;

an adder, a plus-oner, a plus-two-er and a leading zero anticipator (LZA) each coupled to the PKG generator;

a rounding control unit coupled to the LZA;

a multiplexor coupled to each of the adder, the plus-oner, the plus-two-er and the rounding control unit; and

and a normalization shifter coupled to the multiplexor and the LZA.

2. (Original) The significand portion of Claim 1 wherein the multiplier outputs a sum value and a carry value.

3. (Original) The significand portion of Claim 2 wherein the PKG generator computes a propagate value (P), a kill value (K) and a generate value (G) based on the sum value and the carry value.

4. (Original) The significand portion of Claim 3 wherein in parallel the adder adds the sum value and the carry value using P, K, and G; the plus-oner adds the sum value and the carry value using P, K, and G and increments by one; and the plus-two-er adds the sum value and the carry value using P, K, and G and increments by two.

5. (Original) The significand portion of Claim 4 wherein the LZA computes in parallel with the adder, the plus-oner, and the plus-two-er.

6. (Original) The significand portion of Claim 1 wherein the rounding control unit reads a rounding mode from a register in a processor in which the FMAC resides.

7. (Original) The significand portion of Claim 1 wherein the normalization shifter and the rounding control unit each receive a leading zero position indication from the LZA.

8. (Original) The significand portion of Claim 1 wherein the multiplexor produces an output result responsive to the rounding control unit.

9. (Original) A floating point multiply accumulator (FMAC) comprising:

a multiplier;

a propagate, kill, generate generator (PKG generator) to produce a propagate value (P), a kill value (K) and a generate value (G) coupled to the multiplier;

an adder, a plus-oner, a plus-two-er and a leading zero anticipator (LZA) each coupled to the PKG generator in parallel;

a rounding control unit coupled to the LZA and coupled to a multiplexor, the multiplexor outputting a result from one of the adder, the plus-oner, and the plus-two-er responsive to the rounding control unit; and

and a normalization shifter coupled to the multiplexor and the LZA.

a1

10. (Original) The FMAC of Claim 9 wherein the multiplier produces a product of a first floating point number and a second floating point number added to a third floating point number as a sum value and a carry value.

11. (Original) The FMAC of Claim 10 wherein in parallel the adder adds the sum value and the carry value using P, K, and G; the plus-oner adds the sum value and the carry value using P, K, and G and increments by one; and the plus-two-er adds the sum value and the carry value using P, K, and G and increments by two.

12. (Original) The FMAC of Claim 9 wherein the rounding control unit outputs a select signal to the multiplexor based on a rounding mode and the decimal point position.

13. (Original) The FMAC of Claim 9 wherein the normalization shifter normalizes based on the decimal point position.

14. (Original) A floating point multiply accumulator (FMAC) comprising:

a means for multiplying a first significand and a second significand and adding a third significand to produce a sum value and a carry value;

a means for computing a propagate value, a kill value, and a generate value coupled to the means for multiplying;

a first means for adding the sum value to the carry value;

a second means for adding the sum value to the carry value and incrementing by one;

a third means for adding the sum value to the carry value and incrementing by two;

a means for determining a leading zero position, such that the first means for adding, the second means for adding, the third means for adding, and the means for determining are coupled in parallel to the means for computing;

a means for controlling responsive to the means for determining and a rounding mode, the means for controlling further coupled to a means for selecting, the means for selecting outputting a result from one of the first means for adding, the second means for adding, and the third means for adding responsive to the means for controlling; and

and a means for normalizing coupled to the means for selecting and the means for determining.

15. (Original) The FMAC of Claim 14 wherein the means for controlling reads the rounding mode from a register in a processor in which the FMAC resides.

16. (Original) The FMAC of Claim 14 wherein the means for normalizing is responsive to the means for determining.

17. (Original) A method in a floating point multiply accumulator (FMAC) comprising:

receiving a first floating point number, a second floating point number and a third floating point number;

computing a product of the first floating point number and the second floating point number and adding a third floating point number to produce a sum value and a carry value;

computing a propagate value, a kill value and a generate value based on the sum value and the carry value;

simultaneously adding the sum value to the carry value to create a first result, adding the sum value to the carry value and incrementing by one to create a second result, adding the sum value to the carry value and incrementing by two to create a third result, and determining a decimal point position;

selecting one of the first result, the second result and the third result responsive to a rounding mode and the decimal point position as a selected result; and

normalizing the selected result based on the decimal point position.

18. (Original) The method of Claim 17 further comprising;

reading the rounding mode from a register in a processor in which the FMAC resides.

19. (Original) The method of Claim 17 wherein normalizing comprises:

shifting the bits in the selected result.

20. (Original) The method of Claim 17 wherein the propagate value, the kill value and the generate value are used by the adder, the plus-oner and the plus-two-er to compute the first result, the second result and the third result.

21. (Original) A machine readable medium containing instructions which, when executed by a processor, cause a machine to perform operations comprising:

receiving a first floating point number, a second floating point number and a third floating point number;

computing a product of the first floating point number and the second floating point number and adding a third floating point number to produce a sum value and a carry value;

computing a propagate value, a kill value and a generate value based on the sum value and the carry value;

simultaneously adding the sum value to the carry value to create a first result, adding the sum value to the carry value and incrementing by one to create a second result, adding the sum value to the carry value and incrementing by two to create a third result, and determining a decimal point position;

selecting one of the first result, the second result and the third result responsive to a rounding mode and the decimal point position as a selected result; and

normalizing the selected result based on the decimal point position.

22. (Original) The machine readable medium of Claim 21 containing instructions which, when executed by a processor, cause the machine to perform further operations comprising:

reading the rounding mode from a register in a processor in which the FMAC resides.

23. (Original) The machine readable medium of Claim 21 wherein normalizing comprises:

shifting the bits in the selected result.